

Create a Backup

by Sophia Tutorial



WHAT'S COVERED

This tutorial explores using the command line to back up a PostgreSQL database in two parts:

1. Introduction
2. PostgreSQL Backup Tools

1. Introduction

Depending on the choice of database, backing up a database can take various forms. In most cases, there is a command line option and a graphical user interface option that functions by using the command line behind the scenes. These steps are important to save the data and state of the database, so that it can be recovered if there are any issues. There are different types of database backup options, such as full or partial data, or being able to choose to back up structures.

2. PostgreSQL Backup Tools

In PostgreSQL databases, we have the `pg_dump` and `pg_dumpall` tools. These will not work within our web interface tool, as the tool is logged into PostgreSQL already. However, if you have PostgreSQL installed locally, you can test the commands.

The `pg_dump` tool outputs all of the contents of all database objects into a single file. The script dumps contain SQL commands that are in plain-text files that can be used to reconstruct the database back to the state that it was in when the database was backed up.

There are a lot of different parameters that can be used, so we will explore some of the more common options.

Let us first look at a complete command that will back up the `mydb` database to a `mydb.sql` file in the `c:\backups\` folder on a Windows system, using the user `adminrole`:

```
pg_dump -U adminrole -W -F p mydb > c:\backup\mydb.sql
```

Let's break this down:

- The `pg_dump` is the command line tool.
- The `-U adminrole` specifies the user role that will be used to connect to the database. In this case, we are

using the adminrole to login to perform the backup.

- The -W prompts the pg_dump command to prompt for the password on adminrole before it can continue.
- The -F specifies the output file format. In this case, the p stands for plain-text SQL script file.
- Then, we indicate the database that we want to backup, which is mydb.
- The > c: \backup\mydb.sql is the output backup file name and path that we are backing up to. If you don't pass in a path, you can just include the file name, which will output the backup file to the current directory that you are running the command in.

Let us explore some other options:

- The -a option will only dump out the data, but not the schema with the data definitions like the table structure. This approach is only useful if we back up the data in plain text, as it will allow us to export just the data.
- The -s will dump out just the object definitions like the tables, rather than including the data.
- The -c with the lowercase c will clean the database objects by drop statements first before creating them.
- The -t will only dump out specific tables that match the table name that is passed. For example, -t employee would only dump out the employee table.
- The -T with a capital T will dump out all tables other than the ones that are listed.
- The -C with a capital letter will start the output with a command to create the database and reconnect to the created database. This option allows you to avoid having to create the database first.

With the -F option to indicate the format, we looked at p to output the plain-text SQL script. There are other options as well, like "d" for directory, where it will create a directory with one file for each table. There is also "t" for a tar, which will create an archive file similar to a .zip file.

The pg_dumpall command will dump out all of the databases within a server. This is not a commonly used option, as you typically would only want to back up specific databases at a time. The pg_dumpall process will export all of the databases into a single file, so restoring from this can be unreliable. All of the options with pg_dumpall command will be the same as with the pg_dump command. The only difference is that the -W option is not used, as you wouldn't want to have to type the password for each individual database.



SUMMARY

The pgdump and pg_dumpall commands will allow you to backup your database(s) using the command line.

Source: Authored by Vincent Tran